

PATENT APPLICATION

**OMNIPORTLET - DECLARATIVELY PUBLISH DATA IN A PORTAL
WITHOUT CODE**

Inventor(s): Stephen Burns, a citizen of United Kingdom, residing at
801 Kipling Street
Palo Alto, CA 94301

Robin Fisher, a citizen of United Kingdom, residing at
815 University Avenue
Palo Alto, CA 94301

Assignee: Oracle International Corporation
500 Oracle Parkway, M/S 5OP7
Redwood City, CA 94065

Entity: Large

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 415-576-0200

Oracle Reference No.: OID-2003-137-01

OMNIPORTLET - DECLARATIVELY PUBLISH DATA IN A PORTAL WITHOUT CODE

CROSS-REFERENCES TO RELATED APPLICATIONS

5 **[0001]** The present application is a non-provisional of and claims priority from U.S. Provisional Application No. 60/501,492, Attorney Docket No. 21756-003301US, filed September 8, 2003, the entire contents of which are herein incorporated by reference for all purposes.

BACKGROUND OF THE INVENTION

10 **[0002]** The present invention generally relates to web page design and more specifically to methods and apparatus for generating a page to publish data using a declarative specification.

15 **[0003]** A portal is designed to publish information for access by users. In order to publish the information, a web-based page is typically created. A page designer first determines how the web-based page should look. For example, the page designer may determine which data to display and how it should be displayed on the page. The page designer's preferences for the page are then communicated to a page developer, who then creates software code that generates a page to publish the data according to the page designer's preferences.

20 **[0004]** A page designer is typically a non-technical person who does not know how to write software code that will generate the web page. Rather, the page designer has to rely on a page developer to write the software code to generate the page. The page designer has to design the page and communicate with the page developer, who then develops the page and write the software code to generate the page according to the page designer's preferences. Development thus becomes a bottleneck for the page designer.

25 **[0005]** The page developer may attempt to pre-build pages for the portal in order to alleviate the bottleneck. The pre-built pages allow a page designer to lay out the data on the page without writing software code. However, the pre-built pages are created to display a certain subset of data. If the page designer wants to publish data other than the supported

data for the pre-built pages, then a page developer has to create another page. Accordingly, the pre-built pages are not flexible on which data may be displayed in the page.

BRIEF SUMMARY OF THE INVENTION

5 **[0006]** The present invention generally relates to generating a page from a declarative specification. An interface is provided that enables a user to declaratively specify a data source. Also, an interface is provided that allows a user to declaratively specify layout information for data from the data source. A page is then generated based on the declarative specification for the data source and the layout information.

10 **[0007]** In one embodiment, a method for generating a page is provided. The method comprises: determining a data source specification and a layout specification for the page, the data source specification and layout specification being declaratively specified by a user, the data source specification including access information for a data source; retrieving data for the data source based on the data source specification and the access information;
15 determining a layout for the data based on the layout specification; and generating the page using the data and the determined layout.

[0008] In another embodiment, a method for generating a graphical user interface (GUI) is provided. The method comprises: providing a data source interface that enables a user to declaratively specify access information for a data source; determining a declarative
20 specification for the data source; retrieving data for the data source using the access information; providing a layout interface that enables the user to declaratively specify a layout for the determined data; determining a layout specification for the determined data; and generating a graphical user interface based on the declarative specification for the data source and the layout specification for the determined data.

25 **[0009]** In yet another embodiment, a method for declaratively generating a page using an interface is provided. The method comprises: receiving a declarative specification of access information for a data source, wherein the access information is input by a user; retrieving data using the access information for the data source; determining layout information for data from the data source; and generating a page using the retrieved data and the layout
30 information.

[0010] A further understanding of the nature and advantages of the invention herein may be realized by reference of the remaining portions in the specifications and the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Fig. 1 illustrates a portal page that includes one or more portlets according to one embodiment of the present invention;

[0012] Fig. 2 illustrates a simplified flowchart of a method for generating a portlet according to one embodiment of the present invention;

[0013] Fig. 3 depicts a simplified flow chart of a method for specifying a declarative specification of a data source according to one embodiment of the present invention;

[0014] Fig. 4 illustrates a simplified flowchart a method for specifying a declarative specification of a layout according to one embodiment of the present invention;

[0015] Fig. 5 depicts a system for generating a portal page using a declarative specification according to one embodiment of the present invention;

[0016] Figs. 6A - 6E depict interfaces for declaratively specifying data sources according to one embodiment of the present invention;

[0017] Fig. 7 depicts an interface for declaratively specifying filtered preferences according to one embodiment of the present invention; and

[0018] Fig. 8 depicts an interface for declaratively specifying layout options according to one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0019] Embodiments of the present invention will be described as generating portlets for a portal page. In one embodiment, a portlet is a web-based page or window that is configured to display data from a data source. A portal page is any web-based page that may display one or more portlets. Although embodiments of the present invention describe portlets and portals, it will be understood that the techniques described herein may be used to generate any page that is generated using software code, such as a web page or a graphical user interface.

[0020] Fig. 1 illustrates a portal page 100 that includes one or more portlets 102 according to one embodiment of the present invention. In one embodiment, portal page 100 may be a web-based page. Although portal 100 is described as a web page, portal 100 may be any graphical user interface.

5 [0021] Portlets 102 are depicted as portlets 102-1, 102-2, 102-3, 102-4, and 102-5. Each portlet 102 is configured to display data from a data source according to a specified layout. The data source and layout for data from the data source are declaratively specified by a user. Accordingly, in order to generate a portlet 102, the software code does not need to be written. Rather, a user can declaratively specify the data source and layout. A portlet 102 is then
10 generated using the declarative specifications.

[0022] Portal page 100 may include multiple portlets 102. Thus, data from various sources may be published on portal page 100. Portlets 102 may be independent web-based pages. Also, data parameters in different portlets 102 on portal page 100 may be linked together or be dependent on one another. Although portlets 102 are described, it will be understood that
15 any kind of graphical user interface or page may be generated.

[0023] Fig. 2 illustrates a simplified flowchart 200 of a method for generating portlet 102 according to one embodiment of the present invention. In step 202, one or more interfaces are provided for declaratively specifying a data source. In one embodiment, the interfaces are included in a web-based wizard that guides a user through declarative steps that specify the
20 data source.

[0024] In one embodiment, access information for any data source may be provided by a user. The access information may be a link to any data (e.g., a uniform resource locator (URL)), a storage location, or any other information that is needed to access data for the data source. Additionally, as will be described later, a user may specify a data type and also filter
25 data from the data source to specify which data should be displayed in a portlet 102.

[0025] In step 204, a declarative specification of the data source is determined. When a user declaratively specifies the data source in step 202, a declarative specification of the data source is determined from the input information. The specification includes information that is needed to generate a portlet 102 with the specified data.

30 [0026] In step 206, one or more interfaces for specifying a layout for data from the data source are provided. The one or more interfaces allow a user to declaratively specify a layout

for data from the data source. In one embodiment, data is retrieved using the access information specified for the data source and displayed in the one or more interfaces. The user can then declaratively specify how the data should be displayed in portlet 102. For example, layout types, such as a tabular layout, chart layout, news layout, form layout, and bullet layout may be provided for a user to choose from. The user can then choose how to display or layout the data in a portlet 102.

[0027] In one embodiment, the specification of the layout is separated from the specification of the data source. By separating the specification of the data source from the layout, a layout may be changed without having to rebuild a portlet 102. For example, data from the data source may be rendered initially as a tabular layout and then changed to a chart layout by re-specifying the layout. Because the data source specification is separated from the layout specification, the data may then be reformatted for the changed layout type without designing a new page. A portlet 102 is just created with the new layout for data from the data source.

[0028] In step 208, a declarative specification of a layout is determined. When a user declaratively specifies the layout in step 206, a declarative specification of the layout is determined from the input information. The specification of the layout includes information needed to generate a portlet 102 with the specified layout preferences.

[0029] In step 210, a portlet 102 is generated based on the declarative specification of the data source and the layout. Portlet 102 is generated with the information that was declaratively specified. In one embodiment, software code uses the declarative specification to generate portlet 102. For example, a data driver or renderer generates portlet 102 using the information that was declaratively specified. The data driver and outputted interfaces are developed such that information declaratively defined using the interfaces may be interpreted by the data driver to generate portlet 102. In one embodiment, the declarative specification comprises meta-data that is used by the data driver to generate portlet 102. In another embodiment, software code is generated using the declarative specification where the software code generates portlet 102.

[0030] In one embodiment, the user is not restricted to creating portlet 102 with specific data. Rather, any data source may have been declaratively specified in steps 202 and 204. For example, conventionally, a developer may create a page that supports only weather data from a first source. A page designer would then be restricted to creating a page with only

that weather data. However, using embodiments of the present invention, a user may specify access information to any weather data and create a portlet 102 to display the specified weather data.

[0031] Fig. 3 depicts a simplified flow chart 300 of a method for specifying a declarative specification of a data source according to one embodiment of the present invention. In step 302, an interface for declaratively specifying the data source type is provided. In one embodiment, the data types provided include a spreadsheet, structured query language (SQL), extensible markup language (XML), web service, and web page data types. Although the above data types are described, it will be understood that a person skilled in the art will appreciate other data types that may be used.

[0032] Data stored in spreadsheets is one type of data type that is supported. Typically, spreadsheets are a common method of storing small data sets. Data in a spreadsheet is often created as a comma separated value (CSV) file. Embodiments of the present invention are configured to retrieve data from a CSV file or any other file that stores data in a spreadsheet format.

[0033] A SQL data type includes any data stored in a relational database. In one embodiment, JDBCTM drivers are used to access a relational database. For example, SQL statements are generated and used to retrieve data for a specified data source.

[0034] The XML data type allows a user to publish data from an XML data source.

Typically, a URL for an XML data source is provided and data is retrieved from a storage location associated with the URL. The structure of the XML data source may be specified by a user. For example, an XML schema or XML style sheet that describes the structure of the data may be specified. Also, embodiments of the present invention may determine the data structure for the XML data source. In one embodiment, data retrieved from an XML data type is expected in a ROWSET/ROW structure. Although a ROWSET/ROW structure is used, it will be understood that any hierarchical structure may be supported.

[0035] A web service data type is a service that may be accessed over the Internet using standard web protocols, such as SOAP and HTTP. Web service protocols are typically non-platform and non-language specific. For example, a web service may be a streaming data service for stock quotes, a news brief, etc.

[0036] In one embodiment, the web service data type includes the types of document/literal and remote procedural call (RPC)/encoded. Document and RPC are different mechanisms for communicating with web services. It will be understood that other mechanisms to communicate with web services will be appreciated and supported.

5 [0037] In one embodiment, web services data is parsed using the ROWSET/ROW format. Also, a user can specify the format; for example, a user may specify a link to a file (e.g., an XSL or XML schema file) that specifies the format of the web service data type.

[0038] A web page data type includes any information that may be found on a web page. The web page type enables a user to capture data from a web-based application. The data
10 from a web page may be selected and reformatted.

[0039] In step 304, a declarative specification of the data source type is determined. When a user declaratively specifies the data source type in step 302, a declarative specification of the data source type is determined from the input information. The declarative specification includes information that is used to generate a portlet 102 for the data source type.

15 [0040] In step 306, an interface for specifying a data source based on the data source type selected is provided. Depending on the data type selected, the access information required to retrieve data for a data source may be different. For example, for a spreadsheet, a storage location for the spreadsheet or a URL may be provided as access information. Fig. 6A depicts an interface 600 for declaratively specifying a spreadsheet data source according to
20 one embodiment of the present invention. As shown, access information may be specified in an entry 602.

[0041] For a SQL data type, a SQL statement may be provided that will retrieve data from a SQL database from a relational database. Fig. 6B depicts an interface 610 for declaratively specifying a SQL data source. As shown, access information (e.g., a SQL statement) for data
25 in a relational database may be provided in an entry 612. The SQL statement may be used to retrieve data from a data source specified in the SQL statement.

[0042] For an XML data type, a link (e.g., a URL) to the XML data is specified. Also, information that describes how the data is stored may also be specified. For example, an XML style sheet (XSL) or an XML schema that describes the data structure of the data may
30 be specified.

[0043] Fig. 6C depicts an interface 620 for declaratively specifying an XML data source according to one embodiment of the present invention. Access information may be entered in an entry 622 for the XML data source. As shown, an XML URL has been entered in entry 622. Also, an entry 624 is provided that allows a user to enter a link to an XSL filter URL.

5 The XSL filter describes the data structure of the XML data source. Also, an entry 626 is provided for a link to an XML schema that describes the data structure of the XML data source. Entries 624 and 626 may be used if the data are stored in a ROWSET/ROW format. Embodiments of the present invention are configured to retrieve data in that format; however, an XSL or XML schema may be used if data are in a different format than ROWSET/ROW.

10 [0044] For a web services data type, a link (e.g., a URL) to the web service may be specified. A user may also specify an XSL file or XML schema that describes the data structure of data from the web service.

[0045] Fig. 6D depicts an interface 630 for declaratively specifying a web service data source according to one embodiment of the present invention. Access information for the web server data source may be provided in an entry 632. As shown, a URL to a web service has been inputted in entry 632. The URL may be to a web service description language (WSDL) that describes the web service, specifies the methods that are available to be called and the parameters that are expected, and a description of the data. Once the access information is inputted, a Web Service methods button 634 may be selected. Different methods 636 may then be determined for the web service. In one embodiment, the methods provide multiple operations that may be called to execute and return results. The user may select which operation he or she is interested in and the operations are called to retrieve data. Parameters for the method may be provided in an entry 638. Once the user has selected a method (or operation), the user may enter parameters (also referred to as parts) to pass to that method. The WebService's WSDL (Web Services Description Language) is parsed to locate all the methods (or operations) that can be called and the associated parameters (parts) for each method in order to retrieve specified data.

[0046] Also, an XSL link may be inputted in an entry 640 and/or an XML schema may be inputted in an entry 642 that describes the data structure of the data. Entries 640 and 642 may be used if the data are stored in a ROWSET/ROW format. Embodiments of the present invention are configured to retrieve data in that format; however, an XSL or XML schema may be used if data are in a different format than ROWSET/ROW.

[0047] For the web page data type, a portion of a web page may be selected and data from that selection is retrieved. Data from the web page is retrieved and displayed. A user may then select columns of data from the displayed data that a user desires to publish in portlet 102.

5 [0048] Fig. 6E depicts an interface 650 for declaratively specifying a web page data source according to one embodiment of the present invention. When a web page or part of a web page is selected, data from that web page is retrieved and displayed in a window 652. Check boxes 654 are provided for data retrieved. In one embodiment, check boxes 654 are provided for columns of data from the web page. A user may then select the data that is desired using
10 check boxes 654.

[0049] In one embodiment, access to secure data is supported for all data sources described above. Connection information may be specified if a data source is secured. For example, login information (e.g., a username and password) may be needed to access the data source. In this case, a user may provide the information, which is used to access the secure data.
15 Also, data protected by HTTP authentication may be accessed.

[0050] In one embodiment, portlet 102 parameters may also be provided for all the data sources described above. The parameters may be used to specify which data is returned from a data source. For example, the parameters may be used to specify that only data from a "department 10" should be returned. This allows filtering to be done at the data source. In
20 another embodiment, filtering may be done with data received from the data source, as described below.

[0051] Referring back to Fig. 3, in step 308, a declarative specification of the data source is determined. When a user declaratively specifies the data source in step 306, a declarative specification of the data source is determined from the input information. In one
25 embodiment, the access information specified in step 306 is determined.

[0052] In step 310, data is retrieved from the specified data source and an interface is provided for specifying any filtering for the retrieved data. The interface allows a user to filter and/or sort the data that was retrieved. The filtering options provided in this step allow a user to filter data for data retrieved using the access information provided. A user may
30 filter the data on a different level by specifying access information for the data source that only retrieves data that is desired. However, if this is not be done, filtering options for filtering the retrieved data may be used.

[0053] Fig. 7 depicts an interface 700 for declaratively specifying filtered preferences according to one embodiment of the present invention. As shown, conditions may be specified in entries 702 that specify which data should be displayed in portlet 102. The conditions may specify specific data to include or may set conditions that will filter data automatically according to the conditions. For example, filtering may be used to selected data from a column that is less than a certain value. Also, the order of data displayed in portlet 102 may also be specified in entries 704.

[0054] In step 312, a declarative specification is determined for any filtering preferences. When a user declaratively specifies the data source type in step 310, a declarative specification of the data source is determined from the input information. The declarative specification includes information that is used to generate a portlet 102 with the specified filtering preferences.

[0055] In step 314, data from the data source is filtered based on the filtering preferences. Thus, a user has declaratively specified a data source and any access information that is needed to access the data from the data source. Also, the data retrieved is filtered to determine data that is desired by the user.

[0056] In step 316, the data source specification is generated based on the declarative specifications determined in steps 308 and 312. In one embodiment, the data source specification may be information that is used to generate a portlet 102 with data that a user declaratively specified.

[0057] Fig. 4 illustrates a simplified flowchart 400 a method for specifying a declarative specification of a layout according to one embodiment of the present invention. In step 402, an interface for declaratively specifying a layout type is provided. In one embodiment, the layout type may include a tabular, chart, news, form, and bullet layout type. Although the above layout types are described, it will be understood that a person skilled in the art will appreciate other layout types that may be provided.

[0058] Fig. 1 depicts portlets 102 that include the above layout types. For example, portlet 102-1 depicts a tabular layout type; portlet 102-2 depicts a news layout type; portlet 102-3 depicts a bullet layout type; portlet 104-4 depicts a chart layout type; and portlet 102-5 depicts a form layout type.

[0059] In step 404, a declarative specification of the layout type is determined. When a user declaratively specifies the layout type in step 402, a declarative specification of the layout type is determined from the input information. The declarative specification includes information that is used to generate a portlet 102 with the specified layout type.

5 [0060] In step 406, an interface is provided for layout options based on the data retrieved in step 314 of Fig. 3 and the layout type selected in step 402. For each layout type, different layout options may be provided. For example, for a tabular layout, a user may select whether alternating colors for each row should be used. For a news layout, a user may select whether an image (typically the image of the news provider) is shown or news is shown in an
10 automatic scrolling zone. For the bullet layout, a bullet format, such as disk, number, letter, roman numerals, etc., may be provided. For a form layout, whether the form fields are horizontal or vertical may be provided. For the chart layout, the type of chart (e.g., bar, pie, line) and if the data is already grouped, which column it is grouped by, may be provided. Although the above layout options are described, it will be understood that a person skilled in
15 the art will appreciate other layout options that may be provided in the interface for selection by a user. For example, a user may specify whether a column should be displayed in a specific format such as plain text, HTML, as an image, as a button, or as a field. In one example, a user may select that a data source that contains a URL for an image should be displayed as the image.

20 [0061] Also, a user may choose an option that allows a hyperlink to be used for data retrieved. For example, a hyperlink may be provided so that when selected in portlet 102, an action may occur, such as jump to another URL or jump to a page associated with the URL that includes data.

[0062] Fig. 8 depicts an interface 800 for declaratively specifying layout options according
25 to one embodiment of the present invention. As shown, a layout type 802 may be selected to specify the layout type. Depending on the layout type selected, a preview of the data for the selected layout type is shown in entry 804.

[0063] In step 408, a declarative specification for the layout options is determined. When a user declaratively specifies the layout option in step 406, a declarative specification of the
30 layout options is determined from the input information. The declarative specification includes information that is used to generate a portlet 102 with the specified layout options.

[0064] In step 410, a layout specification based on the declarative specifications for the layout type and layout options is generated. In one embodiment, the layout specification may be information that is used to generate a portlet 102 with the layout for data that a user declaratively specified.

5 **[0065]** Fig. 5 depicts a system 500 for generating a portal page using a declarative specification according to one embodiment of the present invention. System 500 includes interfaces 502, a data source engine 504, a layout engine 506, and a portlet instance 512.

[0066] Data source engine 504 is configured to generate interfaces 502 that enable a user to declaratively specify a data source. Once a data source is declaratively specified, data source engine 504 is configured to access data for the data source stored in a database 508.
10 Depending on the data type of the data source and the access information provided, data source engine 504 may use various methods for accessing database 508. For example, a URL may be used to access information for an XML database and an HTTP protocol may be used to retrieve information in a web page, etc.

15 **[0067]** Data source engine 504 includes drivers for retrieving data of any data type using access information as specified through interfaces 502. The drivers are compatible with the interfaces outputted in that the drivers are configured to read information that was declaratively specified using the interfaces. The information is then used to retrieve data.

[0068] When a user declaratively specifies the data source, information for the data source is stored in a portlet instance 512. For example, the data source information may include
20 access information for the data source, its data type, and any filtering information for the data.

[0069] Layout Engine 506 is configured to output interfaces 502 that enable a user to declaratively specify layout information for the data source specified. In one embodiment,
25 data from database 508 is displayed in an interface 502 and the user may select or specify layout options for the data.

[0070] Once the layout information is declaratively specified using interfaces 502, the layout information is stored in a portlet instance 512. The layout information may include the layout type and any layout options specified.

30 **[0071]** Accordingly, portlet instance 512 is declaratively specified by a user. The declarative specification is stored in portlet instance 512 and can be used to generate a portlet

102 in portal 100. The information may be used by software code to generate a portlet 102. In this case, many portlet instances 512 may be created using software code that is configured to generate portlets 102 from the information in each portlet instance 512. For example, as mentioned above, a data driver or renderer generates portlet 102 using the information that
5 was declaratively specified. The data driver and outputted interfaces are developed such that information declaratively defined using the interfaces may be interpreted by the data driver to generate portlet 102. In one embodiment, the declarative specification comprises meta-data that is used by the data driver to generate portlet 102. In another embodiment, software code is generated using the declarative specification where the software code generates portlet 102.

10 **[0072]** In creating portlets 102, data from database 508 is retrieved using the data source information and formatted according to layout information in portlet instance 512. Portlet 102 is then generated and displayed in portal 100.

[0073] Accordingly, embodiments of the present invention enable a portlet to be declaratively specified. A portlet is then generated from the declarative specifications. A
15 data source may be specified in addition to how to lay out data from the data source. Thus, a user can specify the data to be accessed and displayed, and how to lay it out. Previously, a user would have to write software code for the portlet using a programming language. However, using embodiments of the present invention, a user does not have to write software code. Rather, a user may declaratively specify a portlet. Accordingly, time that a user would
20 have to wait for development to build, test, deploy a portlet is avoided. Thus, time to market for a portal page is reduced.

[0074] While the present invention has been described using a particular combination of hardware and software implemented in the form of control logic, it should be recognized that other combinations of hardware and software are also within the scope of the present
25 invention. The present invention may be implemented only in hardware, or only in software, or using combinations thereof.

[0075] The above description is illustrative but not restrictive. Many variations of the invention will become apparent to those skilled in the art upon review of the disclosure. The scope of the invention should, therefore, be determined not with reference to the above
30 description, but instead should be determined with reference to the pending claims along with their full scope of equivalents.